

Hinweise zur MuPAD Syntax

Voraussetzung: Es ist bereits bekannt, wie MuPAD gestartet und ein Notebook (oder: *Arbeitsblatt*) geöffnet wird, interaktive Eingaben, Änderungen und Berechnungen ausgeführt sowie Funktionsgraphen und 3D-Grafiken erstellt werden. Der Umgang mit der MuPAD Benutzungsschnittstelle ist ebenfalls vertraut.

Auf den folgenden Seiten werden Hinweise zur Syntax von MuPAD Eingaben gegeben, demonstriert wie Zuweisungen an Variablen vorgenommen, Funktionsterme definiert und elementare MuPAD Funktionen aufgerufen werden.

An dieser Stelle wird nur auf grundlegende Aspekte der MuPAD Syntax und ihrer Semantik eingegangen. Weitere Information finden Sie in den folgenden Lektionen, im *Handwerkskasten* und den *Arbeitsblättern* sowie in den detaillierten Hilfeseiten der MuPAD Online Dokumentation.

Im folgenden sind Eingaben in der MuPAD Syntax in **rot** und die zugehörigen Ausgaben darunter in **blau** dargestellt.

Was Sie unbedingt beachten sollten:

- MuPAD unterscheidet bei Funktionsnamen und Variablenamen zwischen **Groß- und Kleinschreibung**! Die Funktion factor(...) zum Ausklammern kann also nicht in der Form Factor(...) aufgerufen werden. Sofern Factor(...) nicht anderweitig definiert wurde, fasst MuPAD sie als undefinierte, d.h. abstrakte, Funktion auf, die sich zu sich selbst evaluiert:

```
◦ factor(x^2-1)
```

```
(x-1)·(x+1)
```

```
◦ Factor(x^2-1)
```

```
Factor(x2 - 1)
```

- Sie müssen alle Operatoren, insbesondere das **Multiplikationszeichen *** **explizit angeben**. Der Ausdruck $3x^2-4$ kann also nicht als $3x^2-4$ eingegeben werden. Folgende Eingabezeichen werden in MuPAD verwendet: Addition: '+', Subtraktion: '-', Multiplikation: '*', Division: '/', Potenz: '^' (vor dem Exponenten). Wurzelterme werden durch einen negativen Exponenten definiert. Die Quadratwurzel einer Zahl n kann auch als 'sqrt(n)' eingegeben werden:

```
◦ 3 * x^2 - 4
```

```
3·x2 - 4
```

```
◦ 3 x^2 - 4
```

```
Error: Unexpected 'identifizier' [line 1, col 4]
```

- MuPAD **rechnet standardmäßig algebraisch-exakt**. Numerische Lösungen erhalten Sie nur bei Verwendung von Funktionen der numeric Bibliothek oder wenn Sie Berechnungen im Kontext der Funktion float(...) durchführen. Zum Beispiel: PI bzw. float(PI):

```
◦ solve( x^2-42, x )
```

```
{√42, -√42}
```

```
◦ numeric::solve( x^2-42, x )
```

```
{-6.480740698, 6.480740698}
```

```
◦ float( solve( x^2-42, x ) )
```

```
{-6.480740698, 6.480740698}
```

```
◦ PI = float( PI )
```

```
π = 3.141592654
```

```
◦ E/2 = float( E/2 )
```

```
 $\frac{e}{2} = 1.359140914$ 
```

- MuPAD **rechnet standardmäßig mit 10 signifikanten Stellen**. Über die Variable DIGITS kann die Anzahl der signifikanten Stellen mit denen gerechnet wird jedoch nahezu beliebig erhöht werden. Zum Beispiel: DIGITS:= 1000; float(PI); delete DIGITS. Das letzte Kommando löscht den Inhalt einer Variablen bzw. setzt MuPAD Systemvariablen auf Ihren Ausgangswert zurück:

```
◦ DIGITS:= 1000: // ':' unterdrückt hier die Ausgabe
◦ float(PI)
```

```
3.1415926535897932384626433832795028841971693993751\
058209749445923078164062862089986280348253421170679\
821480865132823066470938446095505822317253594081284\
811174502841027019385211055596446229489549303819644\
288109756659334461284756482337867831652712019091456\
485669234603486104543266482133936072602491412737245\
870066063155881748815209209628292540917153643678925\
903600113305305488204665213841469519415116094330572\
703657595919530921861173819326117931051185480744623\
799627495673518857527248912279381830119491298336733\
624406566430860213949463952247371907021798609437027\
705392171762931767523846748184676694051320005681271\
452635608277857713427577896091736371787214684409012\
249534301465495853710507922796892589235420199561121\
290219608640344181598136297747713099605187072113499\
999983729780499510597317328160963185950244594553469\
083026425223082533446850352619311881710100031378387\
528865875332083814206171776691473035982534904287554\
687311595628638823537875937519577818577805321712268\
06613001927876611195909216420199
```

- `delete DIGITS:`
- **Zuweisungen an Variablen** erfolgen durch Verwendung des Operators `:=`. Im Gegensatz dazu definiert `=` eine Gleichung bzw. dient in Kontrollstrukturen wie `if`, `while`,... als Vergleichsoperator:

◦ `Wert := 42`

42

◦ `Wert := Wert / 6`

7

◦ `Wert = 8`

7=8

◦ `bool(Wert = 8)`

false

◦ `if Wert = 7 then ja else nein end`

ja

- **Mehrere Eingaben in einer Eingaberegion** bzw. Eingabezeile müssen durch ein Semikolon `;` oder einen Doppelpunkt `:` getrennt werden. Letzterer unterdrückt dabei die Ausgabe des berechneten Ergebnisses für die voranstehende Eingabe. Einen Zeilenvorschub (ohne die Eingaberegion zugleich zu evaluieren) erzeugen Sie dabei mit der Tastenkombination **Shift+Return** bzw. **Shift+Enter** bzw. **Umschalttaste+Return** bzw. **Umschalttaste+Enter**:

◦ `n := 5; Liste := [i^2 $ i=1..n]`

5 [1, 4, 9, 16, 25]

◦ `n := 5;
Sequenz := i^2 $ i=1..n`

1, 4, 9, 16, 25

- MuPAD verfügt über **geschützte Variablen bzw. Bezeichner**, wie zum Beispiel `PI` für die Zahl π , `E` für die Eulersche Konstante $e oder `linalg` für die Bibliothek der Funktionen zur Linearen Algebra. Diese können vom Anwender nicht (ohne weiteres) überschrieben werden:$

◦ `PI := 13`

Error: Invalid left-hand side [PI]

◦ `E := 42`

Error: Identifier 'E' is protected [_assign]

◦ `linalg := 666`

Error: Identifier 'linalg' is protected [_assign]

- **Funktionsnamen und Variablennamen** (genau genommen sind Funktionsnamen auch Variablennamen) müssen in MuPAD stets mit einem Buchstaben a-z, A-Z oder dem Sonderzeichen `_` beginnen, gefolgt von weiteren Buchstaben oder Zahlen 0-9. Variablennamen der Art `_add` sollten gemieden werden, da Namen dieses Typ häufig für interne MuPAD Funktionen verwendet werden, z.B. `_plus` als funktionale Schreibweise für den Operator `+`:

◦ `1 + 4 + 9 + 16 + 25`

55

◦ `_plus(k^2 $ k=1..5)`

55

- **Termumformung und Termvereinfachung**: Zur Termumformung stehen unter anderem `expand` (expandieren) und `factor` (faktorisieren) zur Verfügung. Termvereinfachungen erfolgen in MuPAD oft automatisch oder können auch mit der Funktion `simplify` (vereinfachen) durchgeführt werden:

◦ `T := 2 * (x+1)^2 - 18`

$2 \cdot (x+1)^2 - 18$

◦ `expand(T)`

$4 \cdot x + 2 \cdot x^2 - 16$

◦ `factor(%)`

$2 \cdot (x+4) \cdot (x-2)$

◦ `a*b+b*a = x^2*y*x*y`

$2 \cdot a \cdot b = x^3 \cdot y^2$